

Proposta de uma meta-heurística evolucionária híbrida para o problema do caixeiro viajante com quotas, múltiplos passageiros, transporte incompleto e delay de permanência

Prof Dr. Marco César Goldberg
Islame Felipe da Costa Fernandes
Bruno de Castro Honorato Silva

Departamento de Informática e Matemática Aplicada - DIMAp
Universidade Federal do Rio Grande do Norte - UFRN

Natal, 26 setembro de 2017

Proposta inicial

Hibridização

Combinar o genético com o Simulated Annealing

Busca Local

Simulated Annealing atuará na disposição dos vértices da rota (Vizinhança swap)

Formato do cromossomo

Passageiros: 1 2		Passageiros: 3 4 5			Passageiros: 6 7		Passageiros: 8 9		
1		2			3		4		Nós
1		0			1		1		Bonus
1	1	2	-1	2	-1	3	4	4	Embarque
3	4	3	-1	4	-1	1	1	1	Desembarque

NOVA: População inicial

- Metade com LK: procedimento *geraIndividuoLK()*;
- Metade com um algoritmo construtivo baseado no bonus (semelhante ao GRASP) : procedimento *geraIndividuoConstrutivo()*

NOVA: População inicial com LK - procedimento *geralIndividuoLK()*

- Escolhe um subconjunto de vértices $V' \subseteq V$ tal que a soma dos seus bonus seja maior ou igual a K ;
- Toma-se a clique correspondente do conjunto V' e aplica-se LK;
- Inicialmente, liga o bonus em todos os vértices;
- Aplica-se carregamento heurístico
- Cromossomos de tamanhos variados

NOVA: População inicial - Construtivo + Busca local

procedimento *geraIndividuoConstrutivo()*

- Fase construtiva:
 - ▶ Começa no vértice inicial s
 - ▶ Iterativamente, vai adicionando vértices à rota
 - ▶ O vértice adicionado é escolhido numa lista de candidatos restritos (LRC)
 - ▶ Seja u o último vértice da rota em construção. Seja v o vértice tal que $C(u, v)$ seja mínimo. LRC é composta pelos vértices v' tais que $B(v') > B(v)$ e $C(u, v') \leq C(u, v) * (1 + \alpha)$
- Busca local multi-operadores

Operadores

Cruzamento Recombinação SPLIT

Dado dois pais, A e B , determina-se um ponto de corte em cada um. Gera-se dois cromossomos filhos, $F1$ e $F2$, onde $F1$ é formado pelos alelos da primeira parte de A e pelos alelos de B que ainda não estão em $F1$. Gera-se $F2$ analogamente.

Mutação para o bonus

Inverte o bonus de uma cidade, tal que a restrição do bonus mínimo K continue satisfeita. Em seguida, realiza carregamento heurístico

Mutações para cidades (rota)

- Adicionar cidade (cromossomo aumenta de tamanho - Vizinhança push)
- Retirar cidade (cromossomo diminui de tamanho - Vizinhança pop)
- Trocar cidades (cromossomo permanece do mesmo tamanho)

Simulated Annealing

Entrada: sol : uma solução viável, T_0 : temperatura inicial

```
1  $x_k = sol$ ;  
2  $T_k = T_0$ ;  
3 Seja  $L_k$  a quantidade de iterações para temperatura  $T_k$ ;  
4 enquanto critério de parada não for atingido faça  
5     para  $i=1 \dots L_k$  faça  
6         gerar uma solução  $x_{k+1}$  vizinha (swap na rota) de  $x_k$ ;  
7          $\Delta C = C(x_{k+1}) - C(x_k)$ ;  
8         se  $\Delta C \leq 0$  então  
9              $x_k = x_{k+1}$ ;  
10            se  $C(x_k) < C(sol)$  então  
11                 $sol = x_k$ ;  
12            fim  
13        senão  
14            se  $rand[0, 1] < e^{-\frac{\Delta C}{T_k}}$  então  
15                 $x_k = x_{k+1}$   
16            fim  
17        fim  
18    fim  
19    Atualiza  $T_k$  e  $L_k$ ;  
20 fim
```


NOVO: Renovação da população

- Se o ótimo corrente não mudar em 5 iterações consecutivas:
 - ▶ Renova 1/4 da população aplicando o procedimento *geralIndividuoLK()*
 - ▶ Renova 1/4 da população aplicando o procedimento *geralIndividuoConsturivo()*
- O indivíduos substituídos são sorteados com probabilidade uniforme

Parâmetros

- Tamanho da população : fixo em 100 indivíduos
- Taxa de cruzamento: 0.98
- Taxa de inversão do bonus: 0.5
- Taxa de adição de cidades: 0.1
- Taxa de remoção de cidades 0.4
- Taxa de inversão de duas cidades: 0.1
- Quantidade de gerações: 50
- Taxa de invocação do *simulated annealing*: 0.94
- Temperatura inicial (T_0): 6
- Temperatura final: 1
- Quantidade de iterações em T_0 (L_0): 9
- Fator de decaimento da temperatura: 1.3
- Fator de incremento de L_k : 1.2
- $\alpha = 0.4$

Heurística de carregamento

- ① Ordene os passageiros L em ordem decrescente de tarifa a pagar ;
- ② Para cada passageiro $l \in L$:
 - ① se o vértice de embarque de l estiver na rota, tente embarcar l ;
 - ② Dentre os possíveis desembarques de l , tome aquele nó d da rota tal que o desembarque de l em d tenha penalidade mínima
 - ③ Uma vez que o embarque e o desembarque de l foram determinados, verifique novo fitness da solução.
 - ① Se o fitness melhorou, ok!
 - ② Caso contrário, desfaça o embarque de l e volte à solução anterior (sem l)